• While Schleife:

Neben der for-Schleife ist die while-Schleife von großer Bedeutung in C und C++, da diese sich noch vielseitiger als die for-Schleife von einem Programmierer zur Programmsteuerung einsetzen lässt. Wie bei Schleifen üblich, können mit Hilfe einer while-Schleife im Schleifenrumpf enthaltene Anweisungen so oft wiederholt und ausgeführt werden, wie die im Schleifenkopf formulierte Bedingung erfüllt wird.

for Schleife:

```
for (Anfangswert; Prüfe Bedingung; Veränderung des Anfangswertes) {

    Anweisungen
}
```

while Schleife:

```
Anfangswert;

while (Prüfe ob Bedingung erfüllt bzw. auf true oder false) {

    Anweisungen
    Veränderung des Anfangswertes
}
```

- **Beispiel**

main.cpp

Run

Output

```cpp
#include <iostream>
using namespace std;

int main() {
  for (int i=1; i<=10;i++){
      cout << i << "  ";
  }
    return 0;
}
```

```
/tmp/r6nb701F6S.o
1  2  3  4  5  6  7  8  9  10
```

main.cpp

Run

Output

```cpp
#include <iostream>
using namespace std;

int main() {
    int i=1;
  while (i<=10){
      cout << i << "  ";
      i++;
  }
    return 0;
}
```

```
/tmp/r6nb701F6S.o
1  2  3  4  5  6  7  8  9  10
```

main.cpp

Run

```cpp
/*
Ein Programm, um die Kubikzahlen zu finden, die kleiner oder gleich
    1000 ist.
*/

#include <iostream>
#include <cmath>
using namespace std;

int main() {
int x=1;
while (pow(x,3)<=1000) {
    cout << pow(x,3) << "  ";
    x++;
}
    return 0;
}
```

Output

```
/tmp/r6nb7O1F6S.o
1  8  27  64  125  216  343  512  729  1000
```

main.cpp

Run

Output

```cpp
/*
Ein Programm, um die Fakultät mit einer "while"-Schleife zu finden.
*/

#include <iostream>
#include <cmath>
using namespace std;

int main() {
    int n;
    cout << " give me a number"<< endl;
    cin >>n;

int x=1;
int fact=1;
while (x<=n) {
   fact=fact*x;
    x++;
}
cout << n << "! ist " << fact <<endl;
    return 0;
}
```

```
/tmp/r6nb701F6S.o
give me a number
4
4! ist 24
```

do-while-Schleife :

Die "do-while"-Schleife in C++ ist eine Schleifenstruktur, die es ermöglicht, einen bestimmten Codeblock wiederholt auszuführen, solange eine bestimmte Bedingung erfüllt ist.

Der Unterschied zur "while"-Schleife besteht darin, dass die Bedingung am Ende der Schleife überprüft wird, sodass der Codeblock mindestens einmal ausgeführt wird, auch wenn die Bedingung von Anfang an falsch ist.

cpp

```cpp
do {
    // Codeblock, der wiederholt ausgeführt wird
} while (Bedingung);
```

main.cpp

Run

Output

```cpp
/*
Ein Programm, um die Fakultät mit einer "do - while"-Schleife zu finden.
*/

#include <iostream>
#include <cmath>
using namespace std;

int main() {
    int n;
    cout << " give me a number"<< endl;
    cin >>n;

int x=1;
int fact=1;
do {
   fact=fact*x;
    x++;
}
while (x<=n);
cout << n << "! ist " << fact <<endl;
    return 0;
}
```

```
/tmp/r6nb701F6S.o
give me a number
4
4! ist 24
```

"switch-case"-Anweisung :

Die "switch-case"-Anweisung in C++ ermöglicht die Ausführung von unterschiedlichen Codeabschnitten basierend auf dem Wert einer Variablen oder einem Ausdruck. Sie ist eine alternative Methode zum Einsatz mehrerer "if-else"-Anweisungen, wenn verschiedene Bedingungen getestet werden müssen.

- Der Ausdruck wird ausgewertet, und der Programmfluss springt zum passenden "case"-Zweig, dessen Wert mit dem Wert des Ausdrucks übereinstimmt. Wenn kein passender "case"-Zweig gefunden wird, wird der Codeblock innerhalb des "default"-Zweigs ausgeführt.
- **Der optionale "break"-Schlüsselwort wird verwendet, um den Programmfluss zu beenden und aus der "switch-case"-Anweisung herauszuspringen. Wenn kein "break"-Schlüsselwort verwendet wird, führt der Programmfluss direkt in den nächsten "case"-Zweig über und alle folgenden Codeblöcke werden ebenfalls ausgeführt, bis ein "break"-Schlüsselwort erreicht wird.**

```
switch(Ausdruck) {
    case Wert1:
        Anweisungsblock1
        break;
    case Wert2
        Anweisungsblock2
        break;
    default:
        Anweisungsblock3
}
```

main.cpp

Run

Output

```cpp
#include <iostream>
using namespace std;

int main() {
    int vorwahl;
    cout << " vorwahl ist ..."<< endl;
    cin >>vorwahl;
     switch (vorwahl) {
         case 49:
         cout << " Germany" <<endl;
         break;
         case 91:
         cout << " India" <<endl;
         break;
         case 212:
         cout << " Marokko " <<endl;
         break;
         case 970:
         cout << " Palestine " <<endl;
         break;
         case 380:
         cout << " Ukraine " <<endl;
         break;
         default:
         cout << " Ich weiß es nicht..";
         break;
     }
    return 0;
}
```

```
/tmp/r6nb701F6S.o
vorwahl ist ...
49
Germany
```